

Windows Server

Dave Thompson
Vice President
Windows Server

Windows Server Business

- ◆ Knowledge Worker Infrastructure
 - ❖ Collaboration services platform
 - ❖ Deployment from grass roots to IT
 - ❖ The platform for Office
- ◆ Application Platform
 - ❖ Develop, deploy, operate at any scale
- ◆ IT infrastructure
 - ❖ Secure, global web services
 - ❖ Federated: enterprises, consumers

Windows Server Business

- ◆ Knowledge Worker Infrastructure
 - ❖ Collaboration services platform
 - ❖ Deployment from grass roots to IT
 - ❖ The platform for Office
- ◆ Application Platform
 - ❖ Develop, deploy, operate at any scale
- ◆ IT infrastructure
 - ❖ Secure, global web services
 - ❖ Federated: enterprises, consumers

Themes

- ◆ Absolutely secure, securable
- ◆ Highly reliable, available
- ◆ Simple to deploy and operate
- ◆ End-to-end experience
- ◆ Self-managing, and manageable
- ◆ Any scale

Windows Server Business

- ◆ Knowledge Worker Infrastructure
 - ❖ Collaboration services platform
 - ❖ Deployment from grass roots to IT
 - ❖ The platform for Office
- ◆ Application Platform
 - ❖ Develop, deploy, operate at any scale
- ◆ IT infrastructure
 - ❖ Secure, global web services
 - ❖ Federated: enterprises, consumers

Releases

- ◆ Q3-CY02: .NET Server release
- ◆ CY04: Longhorn release
- ◆ Interim technology/properate
 - ❖ BIG, MMS, Patch fence
 - , and manageable

Windows .NET Server

- ◆ Great application platform
- ◆ Infrastructure for enterprises
- ◆ Secure
- ◆ Easy for “unmanaged” scenarios

Longhorn+: Technologies

- ◆ Platform Security
- ◆ Application Platform
- ◆ Security Infrastructure
- ◆ Directory
- ◆ Management
- ◆ Knowledge Worker Infrastructure
- ◆ File services and storage
- ◆ Testing

Secure Windows

- ◆ Secure boot
- ◆ Run as restricted user
- ◆ Winlogon redesign
- ◆ Anti-virus: common infrastructure
- ◆ System restore
- ◆ Secure coding tools
- ◆ Secure system roles

Security: Infrastructure

◆ Federation

- ❖ Trust management: Passport brokering, PK cross-forest trusts
- ❖ Enterprise (AD/Kerb), consumer

◆ Auditing

- ❖ non-repudiable, scalable collection

◆ Authorization

- ❖ Roles-based access control

Application Platform

- ◆ **Scale-out focus:** the Rack is the computer
- ◆ **BIG V1:** multiple machine scripting, imaging
- ◆ **BIG V2:** dynamic policy, virtualization
- ◆ **Clustering**
 - **Common services**
 - **Challenge:** get ahead of the system features

Security: Infrastructure

◆ Federation

- Trust management: Passport brokering, PK cross-forest trusts

- Enterprise (AD/Kerb), consumer

◆ Auditing

- non-repudiable, scalable collection

◆ Authorization

- Roles-based access control

Directory

Active Directory

Application mode for simplicity in external scenarios

MMS 3.0

- Integration point for enterprise data

- SQL store

- XML, MS scripting languages

- Polyarchy: automatic, flexible data viewing/access

- Authoritative mastering

- Integrated services support

Storage Infrastructure

- Device discovery/alloc
- Block storage virtualization (VDS)
- Topology configuration and fault detection
- Shadow copy transport
- Interconnect independence

Windows NET Server

Windows "Longhorn"

Provided by Vendor

Volume
Shadow
Copy
Service

Destination
Primary path

Virtual
Disk
Service

RAID disk
Secondary path

Hyper-V
Storage
Manager

tape and
optical media
storage operations

Fabric Virtualization Service

(Device Access, SAN Path,
Protocol Abstraction)

Host

Host

Host

PS Filters

File System

Volume Snapshot

Volume Management

Host

Host

Host

Class

Host

Host

Port

Host

Host

Miniport

File System Investments

Longhorn Storage

● New filesystem features:

- Extensible filesystem metadata
- Table-based and XML programming model
- SQL, XPath, and XQuery support
- Property promotion and demotion
- ◆ Windows shell integration

MailStation

NET My Services

Storage Infrastructure

- Device discovery/alloc
- Block storage virtualization (VDS)
- Topology configuration and fault detection
- Shadow copy transport
- Interconnect independence

Windows NET Server

Windows "Longhorn"

Provided by Vendor

Volume Shadow Copy Service

Read-Only
Write-Only

Virtual Disk Service

RAID disk services
File services

Removable Storage Manager

tape and optical used as storage technology

Fabric Virtualization Services

(Device Access, SAN Paths, Protocol Abstraction)

with

File

Volume

File

File

File Systems

Volume Snapshot

Volume Management

File

File

File

Class

File

File

Port

File

File

Miniport

Storage Infrastructure

- Device discovery/alloca
- Block storage virtualization (VDS)
- Topology configuration and fault detection
- Shadow copy transport
- Interconnect independence

Windows NET Server

Windows "Longhorn"

Provided by Vendor

Volume
Shadow
Copy
Service

Disables
Private
Pages

Virtual
Disk
Service

RAID disk
Access
Private
Pages

Removable
Storage
Manager

Tape and
optical media
Virtualization

Fabric Virtualization Service

(Device Access, SAN Policy,
Protocol Abstraction)

Web

File Systems

Volume Snapshot

Volume Management

File System

File

File

File

Class

File System

File System

Port

File System

File System

File System

Miniport

File System Investments

Longhorn Storage

◆ New filesystem features:

- ◆ Extensible filesystem metadata
- ◆ Table-based and XML programming model
- ◆ SQL, XPath, and XQuery support
- ◆ Property promotion and demotion
- ◆ Windows shell integration

Longhorn Information Services

- Collaboration base platform for Office.NET
- Document collaboration environment
 - Natural organization properties
 - Sharing: versioning, notifications
 - Digital Rights Mgt for documents
 - Realtime Communication
- Hailstorm, Longhorn Storage

Hall Storm

NET My Services



Systems Management

Focus on Fundamentals

Goal: Keep the system in a good state

Each individual system component does its best to stay in a good state

Management deals with

- ◆ adding/removing components
- ◆ detecting/correcting bad states, e.g. due to
 - components that aren't smart enough
 - component dependencies/interactions

Management Situation

- ◆ Few components define their state and how to change it, let alone "good state"
 - ▶ A component that can only change state via a GUI isn't manageable
- ◆ More/smaller components, hence more complex dependencies
- ◆ Many components designed with small machine or single user assumptions

Progress, But Not Enough

Progress examples:

- Fusion — side-by-side dlls (Win2K, XP)
- System Restore (WinMe, XP)
- Headless (.NET Server)
- Command line (.NET Server)
- Windows update
- MOM, AppCenter

But relative to progress made in performance and reliability, our manageability is not improving

Why will it be different?

Drivers and Value

- ◆ **Driver:** There's no security without manageability:
 - ◆ efficient SW distribution to fix bugs quickly
 - ◆ effective configuration mgt to reduce errors
 - ◆ effective configuration mgt to reduce product surface area
- ◆ **Value:** Tools guide and offer immediate return on infrastructure investments

Infrastructure + Tools Alone Don't Make a System Manageable

- The box must include the right set of in-the-box management tools
- MS and ISV apps (services, components) must be manageable via these tools
- We'll use *Guides* to wrap manageable apps to drive improvement
 - All infrastructure and tools improvements must be justified by their impact on the *Guide*
 - Do they result in savings we can explain to ISVs?
 - Manageability sign-off criteria and metrics for Windows components will derive from *Guides*

Example: Instrumentation vs Monitoring

Instrumentation (includes event log) is not admin usable

Monitoring systems make instrumentation usable

But no monitoring system has a big share

most Windows servers are unmonitored

the rest: divided between Thwack, Unicenter, HP, MOM

So ISVs don't invest in monitoring rules

So ISV instrumentation is spotty

Conclusion: Windows will include a monitoring service

Based on the needs of the monitoring service, we
invest in instrumentation infrastructure

Management Infrastructure

- ◆ **Software Distribution**
 - ◆ Windows Update, all types of software
- ◆ **Configuration/State**
 - ◆ Consistent state
- ◆ **Monitoring**
 - ◆ Eventing, dynamic state, rules
- ◆ **UI Platform**
 - ◆ Composability, schema driven UI, cmd shell
- ◆ **New services preserve legacy investments**
 - ◆ Eg. New config api, store w/providers for registry
- ◆ **Guide to writing manageable apps**
 - ◆ Full scriptability, new infrastructure

Testing

- ◆ Testing is killing us
 - ❖ Breadth, complexity of the system, infinite scenarios
- ◆ Windows Test Architecture
 - ❖ Modelling: state model drives test case generation
 - ❖ Testability: tools to analyze code for testability
 - ❖ Code "injection": structured testing of failure cases ("Magellan")
 - ❖ Test Lab Management
- ◆ Drill down: DaveY

Server Organization

- ◆ Core Server: Eugene Ho
- ◆ Testing: David Yee
- ◆ Application Platform: Bill Laing
- ◆ File Services: Ben Fathi
- ◆ Hailstorm: Eric Lockard
- ◆ Mgmt Infrastructure: Bharat Shah
- ◆ Directory: Chuck Chan
- ◆ Security: Doug Bayer
- ◆ IT Mgt Tools: Donna Liu
- ◆ Terminal Services: Peter Bergler

Where can you help?

- ◆ Fundamental challenges for customers
 - ❖ Security, reliability, manageability
- ◆ Innovation and alignment for differentiation
 - ❖ Storage and information services
- ◆ Fundamental engineering challenges
 - ❖ testability